

Claim Validity Across Transformation

1. More	2
2. When More Becomes Different	3
3. Claims Can Change Without Looking Different	5
4. Different Claims Need Different Bases	6
5. Claims Accumulate Too	8
6. Case I: The Superapp	11
7. Case II: Superintelligence	13
8. Recursive Capability	15
9. The Moving Horizon	17
10. Continuation Is Not Completion	19
11. Claim Reach and Validity Reach	21
12. Separation Without Fragmentation	23
13. What Accumulation Does Not Decide	25
14. When the Process Changes the Problem	27
15. Conclusion: When More Becomes Different	30
Resources	30
Theoretical Context	31

1. More

Technological progress is often described through accumulation. Systems become faster, larger, more accurate, more connected or more capable. Applications integrate additional functions, models perform better across more tasks, and infrastructures absorb responsibilities that were previously distributed across separate components.

Much of this development is real and measurable. More resources can improve an existing property, and sufficiently large changes can also contribute to genuinely new properties. A system does not have to remain the same kind of system merely because its development began quantitatively. The difficulty lies elsewhere. Continued improvement does not by itself tell us what kind of transformation will eventually occur. A measurable trajectory can establish that something is changing without establishing every property later attributed to the changed system.

Technological language often moves faster than the basis behind it. A system that performs better on more tasks may be described as increasingly general. An application that integrates more functions may be described as increasingly infrastructural. Such descriptions can be useful, and they may eventually become entirely appropriate. But the change in language introduces a new question: what has actually changed in the system, and what supports the broader description?

A simple physical example shows why this is not a rejection of qualitative transformation. Water can undergo a genuine change of state when relevant conditions are reached. The important point is not that every technological transition must possess an equivalent numerical threshold. It is that the new state is identified through properties that differ from those used to describe the preceding state.

Other transformations are more distributed. A capability may emerge through interactions between several components rather than at one identifiable numerical point. A relation can change without the components themselves changing. A system can also become more capable because it becomes better at obtaining information, selecting tools or interacting with its environment rather than merely because it contains more information internally.

The same distinction appears in knowledge. Increasing computational capability can make inference extraordinarily powerful. It can also help a system recognize when its current information is insufficient and when another observation, measurement or interaction is required. Greater intelligence can therefore extend what a system is able to establish without requiring inference and external state to become the same thing.

This becomes more important when learning is recursive. What has already been learned can influence which observation or interaction occurs next, and the result of that interaction can change the next state of learning. Childhood language acquisition provides an intuitive example of this structure: the relevant point is not simply how much information a learner possesses, but how continuing interaction participates in producing what becomes learnable next.^{[1][2]}

Continuation introduces another problem. A process can keep improving without that fact alone establishing every proposed form of completion. How far the development has progressed and what endpoint it establishes are different questions. Claims can therefore expand in several ways: they can cover a different property, a different relation, a different level of a system or a wider set of conditions. The basis that supports the earlier claim does not automatically expand in the same way.

This is where the distinction between Claim Reach and Validity Reach becomes useful. They are not a mathematical scoring system and do not reduce different forms of evidence to one scale. They describe two boundaries: how far a statement extends, and how far the basis for that statement supports it. In its shortest form:

Claim Reach $\leq$ Validity Reach

For now, the relation provides orientation. Its meaning becomes clearer through the transitions by which claims grow, properties change and new evidence becomes available. The two larger cases provide deliberately different tests. The Superapp begins with the accumulation of functions and asks what happens when integration becomes architecture. Superintelligence begins with the accumulation and interaction of capabilities and asks what happens when greater capability becomes a claim about knowledge. Both begin with real forms of more. The question is what that more actually establishes.

2. When More Becomes Different

Quantitative improvement and qualitative transformation are not opposites. More of an existing capability can contribute to a state in which a new property becomes relevant. What matters is not whether quantitative development can produce qualitative change, but what establishes that the transition has occurred.

2.1 A Threshold Is Not Always a Number

Some transitions have identifiable numerical conditions. The change of state of water is an obvious example: under specified conditions, a measurable change is associated with properties that justify describing a different state.

Many technological properties do not have an equivalent single threshold. Resilience, interoperability, autonomy or generality can depend on several components, relations and conditions at once. Their absence of a universal numerical boundary does not make the transition unreal. It means that the property has to be established through the structure in which it appears rather than inferred from an arbitrary amount of accumulation.

2.2 Emergence Does Not Remove the Question

A property can genuinely emerge at system level even when it cannot be meaningfully assigned to any individual component. Coordination, redundancy or interaction can produce behavior that does not exist in the isolated parts. Calling a property emergent explains why examining the components individually may be insufficient, but it does not establish that an expected property exists. The property still has to be identifiable in the resulting system and under the conditions in which it appears. Emergence therefore expands the ways in which a transition can occur; it does not remove the need to establish the transition.

2.3 Structure Can Matter More Than Amount

Two systems can contain the same number of functions and still possess very different properties because those functions are related differently. Consider two platforms that each provide identity, payment, communication, storage and access control. In one, the functions remain separately replaceable and communicate through explicit interfaces. In the other, all five depend on the same identity state, data model and authorization mechanism. The number of functions is identical, but the dependency structure is not.

The same principle applies to information systems. Two systems may have access to comparable information while differing substantially in how observations, records, inference and contextual interpretation remain distinguishable. In both cases, amount alone is insufficient because the relevant property resides partly in structure.

2.4 New Properties Can Depend on Relations

Some properties exist only through relations. Interoperability is not a property of one component considered in isolation; it concerns how components can interact. Substitutability depends on whether one function can be replaced without forcing unrelated changes elsewhere. Prediction concerns a relation between available information and an unresolved state. Verification concerns a relation between a claim and the conditions under which it can be established.

Increasing the capability of an individual component therefore does not automatically improve every property that depends on its relations. A stronger component can participate in a weak architecture, while comparatively limited components can produce robust system behavior through well-structured relations. The transition can therefore occur not only inside an object but between objects, states or responsibilities.

2.5 The Same Change Can Improve One Property and Weaken Another

Technological improvement is rarely one-dimensional. A change that improves one property can simultaneously create a new dependency elsewhere. Deeper integration can reduce friction while increasing the number of functions affected by a shared failure. Automation can remove manual steps while making operation more dependent on a particular infrastructure. Centralization can improve coordination while reducing substitutability, or it can improve resilience when it replaces several inconsistent mechanisms with one more robust implementation.

None of these outcomes follows from integration, automation or centralization by definition. They depend on the resulting structure. This is why describing a system as simply “better” can hide the relevant transition: the improvement may be real while the properties affected by it move in different directions.

2.6 Change Can Also Alter the Conditions

A claim can become broader even when the underlying technical object has not changed. A model tested in one environment can later be described without the original environmental qualification. A component verified under specified assumptions can be discussed as though the result applied independently of those assumptions. A service that has operated reliably within one infrastructure can be described as generally reliable.

In each case, the system may be identical. What changed is the set of conditions carried by the statement. One qualification has disappeared, but the claim has become larger. Conditions therefore belong to the transition whenever removing or changing them alters what is being asserted.

2.7 The Transition Is the Relevant Unit

The preceding cases describe different forms of change. An amount can increase, a new property can emerge, a relation can change, an established result can move between levels, or the conditions attached to a claim can become broader. These changes should not be treated as interchangeable. Each transition raises a different question about what has actually been established.

The useful unit of analysis is therefore not simply the earlier system and the later system, but the transition between them. What property changed? Which relation changed? Did the level of the claim change? Did its conditions change? And what basis supports the statement after that movement?

This shifts the problem from asking whether technology has progressed to asking what travelled with the claim as it progressed. The next step is to examine the claim itself.

3. Claims Can Change Without Looking Different

A technological claim can change even when its wording changes very little. The same term may continue to be used while the property, relation, level or conditions attached to it have expanded. This makes claim transformation difficult to notice. The transition is not always marked by a new concept; sometimes an existing concept simply begins to carry more.

3.1 A Property Can Broaden

Consider a system described as capable because it performs well on a defined set of tasks. If the same description is later used across a wider range of tasks, the word may remain unchanged while the claim has expanded. The broader claim may eventually be justified, but success within the original set does not by itself establish success within the expanded one. Something has changed in what the statement covers.

The same pattern appears outside AI. Reliability demonstrated for one component, workflow or operating environment can become a broader claim about a service or infrastructure. The transition occurs in the scope of the property.

3.2 A Claim Can Move Between Levels

A result established at one level of a system can also be transferred to another. A component may be secure under specified assumptions without establishing the security of the complete system in which it operates. A model may perform reliably on an evaluation without establishing the reliability of an application that surrounds it. An individual service may remain replaceable while the architecture as a whole becomes increasingly difficult to substitute.

These are not contradictions. The relevant properties exist at different levels. Moving from component to system, task to capability or service to architecture therefore changes the claim even when the underlying result remains valid.

3.3 A Claim Can Change Its Relation

Some statements are defined partly by what they relate. A prediction relates available information to an unresolved state. An observation relates a measurement or interaction to a state made accessible through that process. Verification relates a claim to specified conditions or procedures. Substitutability relates a component to an architecture in which replacement is attempted.

A change in one side of the relation can therefore change the claim without changing the central noun or verb. “Verified,” “observed,” “replaceable” or “predictive” only have useful meaning when the relevant relation remains recoverable. Stronger performance in one relation cannot automatically be transferred to another because the new relation requires a basis appropriate to what is now being asserted.

3.4 Conditions Can Disappear Quietly

Conditions are particularly easy to lose because they often appear linguistically secondary even when they are epistemically essential. “Reliable under these operating conditions” can become “reliable.” “Verified against this specification” can become “verified.” “Generalizes across these evaluations” can become “generalizes.”

Nothing in the underlying result has to be false for this to happen. The original claim may remain entirely valid. What changes is that the qualification defining its boundary is no longer carried forward, and the resulting statement reaches further than the statement from which it originated.

3.5 A Claim Can Change Category

Some transitions are larger than ordinary broadening. A claim can move from one category to another. Measured performance can become a claim about general capability. Functional integration can become a claim about resilience. Successful prediction can become a claim about knowledge of an external state. Continued improvement can become a claim about an eventual endpoint.

These transitions may prove justified. But the later claim is not merely a larger version of the earlier one. It introduces a property or relation that must itself be established. This is one of the places where “more” can become “different” in language before the difference has been established in the system.

3.6 A Broader Claim Can Later Become Established

These boundaries are not permanent. A claim that initially exceeds its basis can later acquire direct support. A broader capability can be tested, a system-level property evaluated at system level, or an architectural dependency exposed through operation, replacement or failure. The broader claim then acquires a basis of its own rather than depending only on the transition that first suggested it.

This separates caution from prohibition. Broader claims can be proposed, explored and eventually established; their current status simply has to remain visible while that process is underway.

3.7 The Wording Is Not the Boundary

Natural language does not reliably preserve these distinctions on its own. Terms such as *general*, *autonomous*, *integrated*, *secure*, *intelligent* or *verified* can remain stable while the claims carried by them expand substantially.

The boundary of a claim therefore cannot be inferred from the familiarity of its vocabulary. It has to be recovered from what property is asserted, at what level, in which relation and under which conditions. Once the claim changes, the next issue is the basis capable of supporting what it now says.

4. Different Claims Need Different Bases

Once a claim changes, the issue is no longer only how far it extends, but what kind of basis can support what it now says. There is no single form of support appropriate to every technological claim. Observation, experiment, formal verification, operational history and inference establish different things because they connect a statement to the world, a procedure or a defined system in different ways.

The relevant distinction is therefore not between stronger and weaker evidence in the abstract. It is between a claim and a basis suited to establishing that claim.

4.1 Observation Establishes Through Access

Observation provides access to a state through some form of measurement, sensing or interaction. What it establishes depends on the observation process itself: what was measured, under which conditions, with what instrument and with what uncertainty.

This does not make observation inherently superior to other forms of support. An observation can be noisy, incomplete or dependent on interpretation. Its distinctive role is that the claim is connected to a state through an observational relation. A broader statement requires that this relation actually reaches what the statement covers.

4.2 Experiment Establishes Under Constructed Conditions

Experiments create conditions under which a property, relation or effect can be examined deliberately. Their strength lies partly in this control. The resulting claim therefore includes the experimental conditions. Repetition across different conditions can extend what the result supports, but the initial experiment does not automatically establish the property independently of those conditions.

When a result moves from a controlled setting into a broader technological claim, the experiment may remain entirely valid while the later statement reaches beyond what was tested.

4.3 Formal Verification Establishes Within a Defined Structure

Formal verification can provide exceptionally strong results when the relevant system, specification and assumptions are sufficiently defined. Within that structure, a property may be established deductively rather than inferred from repeated observation. Its boundary follows from the same strength: a proof about a model, specification or defined system does not automatically establish every empirical property of the environment in which that system later operates.

This is not a limitation peculiar to formal methods. It reflects the more general principle that a basis is powerful where its relation to the claim is defined.

4.4 Operational History Establishes Something Different Again

A system that has operated successfully over time acquires a form of support that cannot be reproduced by specification alone. Continued operation can expose conditions, interactions and failure modes that were not fully represented in advance.

Operational history can therefore extend confidence in claims about actual behavior. But successful history does not automatically establish behavior under every future condition, nor does survival within one dependency structure establish independence from that structure. Time can enlarge the basis without making it universal.

4.5 Inference Can Extend Beyond What Is Directly Available

Inference is indispensable because many useful claims concern states, causes or future conditions that are not directly available to observation at the moment the claim is made. Its legitimacy does not depend on pretending that the inferred state has already been observed. It

depends on the relation between available information, the inference process and the claim being made.

A prediction can therefore be well supported without being an observation of the future event. A diagnosis can be strongly supported without every relevant internal state being directly visible. A model can establish a relation with high confidence without converting that relation into a different kind of access. The distinction preserves the strength of inference rather than diminishing it.

4.6 Several Bases Can Support One Claim

Technological claims often depend on more than one form of support. A system may combine formal verification of specified properties, experimental testing, operational history and observation of actual behavior. These bases do not have to compete for the status of the single strongest form of evidence; they can establish different parts of the same broader claim.

What matters is whether the combination covers the statement being made. Several individually valid bases do not automatically support a conclusion that extends beyond all of them, but their explicit composition can establish more than any one could establish alone.

4.7 Uncertainty Does Not Make a Claim Invalid

A valid claim does not require complete certainty. Measurement uncertainty, probabilistic inference, incomplete observation and bounded testing can all remain visible without making the resulting statement useless. Validity depends on whether the uncertainty remains represented within the claim rather than being silently removed from it.

This is especially important for technological development, where many meaningful statements concern systems that are still changing, environments that cannot be exhaustively tested or future states that remain unresolved. Preserving uncertainty can be part of preserving validity.

4.8 The Basis Must Follow the Claim

Different claims therefore require different relations to what supports them. A formal result, an observation, an experiment, an inference and operational history can all be valid without being interchangeable.

As a claim moves between properties, levels, relations or conditions, the relevant basis has to move with it or be supplemented by another one. Otherwise the earlier result may remain valid while the later statement reaches beyond it. This does not require a universal hierarchy of evidence. It requires something simpler: **the basis must be appropriate to the claim it is being asked to support.**

The difficulty grows when several individually plausible transitions are connected into a longer argument. Each step can appear reasonable while the conclusion gradually moves beyond the basis from which the chain began.

5. Claims Accumulate Too

Technological systems accumulate capabilities, functions and evidence. Claims can accumulate in a similar way. One established result supports another statement, that statement becomes part of a broader description, and the broader description then supports the next conclusion. Each individual transition may appear reasonable, but the conditions attached to

the earlier result do not always travel through the entire chain. A claim can therefore move further than any single step suggests.

5.1 A Claim Chain Can Be Individually Plausible

Consider a simplified sequence:

A system performs well on a defined set of tasks.
Therefore, it is capable across that set.

Its performance improves and the set expands.
Therefore, it is becoming more general.

The trajectory continues.
Therefore, it will eventually possess a qualitatively different form of general capability.

The early transitions may be well supported. The final statement may also prove correct, but its validity does not follow merely from the fact that each preceding statement sounded plausible. The chain has changed category: measured performance became a broader capability claim, and continued improvement became a claim about an endpoint.

5.2 A Trajectory Establishes Direction, Not Its Complete Destination

A trajectory can provide strong evidence about direction. Repeated improvement across evaluations can establish that a system is becoming more capable by those measures. Increasing integration can establish that an application is absorbing more functions. What the trajectory does not establish by itself is the complete nature of its destination.

This does not make extrapolation illegitimate. Extrapolation is often useful precisely because current evidence can support expectations about states that have not yet been reached. The status of that expectation, however, differs from the status of what has already been established. Direction and destination can therefore belong to the same argument without being the same claim.

5.3 Intermediate Claims Can Become Premises

Once a broader description becomes familiar, it can begin to function as a premise for later reasoning. A system described as increasingly general may then be discussed in terms of what a generally capable system would know. An application described as increasingly integrated may then be discussed as though infrastructural independence, resilience or substitutability had already been established.

The intermediate claim no longer appears as a transition requiring support; it becomes part of the background. This is how a claim chain can extend quietly. The later argument may be internally coherent while depending on an earlier expansion that was never separately established.

5.4 The Missing Step Can Be Small in Language and Large in Meaning

Many consequential transitions require only a small linguistic change:

“Performs well across more tasks” becomes “general.”
“Integrates more functions” becomes “independent infrastructure.”
“Predicts with increasing accuracy” becomes “knows.”
“Continues to improve” becomes “will reach completion.”

The language becomes shorter while the claim becomes larger. This compression is useful in ordinary communication, but analytically it can hide the point at which a new property, relation or condition entered the argument.

5.5 Conditions Can Be Lost Across Several Valid Steps

The problem becomes harder to notice when no individual step is obviously false. A component can be verified under specified assumptions. The verified component can then operate successfully within a larger system. The system can accumulate a strong operational history. A later description may finally present the system as generally verified or reliable without carrying the assumptions, environment or dependency structure through the chain.

Every intermediate statement may contain something true. The final statement can still exceed what the combined chain establishes. The issue is not that conditions must never change; it is that their removal is itself a transition.

5.6 Claim Chains Can Also Strengthen Validity

Accumulation is not inherently a source of drift. A sequence of observations can extend an empirical basis. Independent experiments can test a property under broader conditions. Operational history can expose interactions that were absent from controlled evaluation. Formal analysis can establish structural properties that empirical testing alone could not guarantee.

A chain becomes stronger when later steps add the basis required by the broader claim rather than merely inheriting the language of earlier ones. Accumulation can therefore expand validity as well as claim scope.

5.7 A Chain Can Be Repaired

A claim chain that has moved too far does not have to be discarded. The missing transition can be identified and tested directly. A system-level property can be evaluated at system level. A broader capability can be tested across the additional conditions it claims to cover. A dependency can be exposed through replacement, failure or operation. A predicted state can later become observable.

The broader claim then acquires the missing basis instead of relying on the trajectory that first suggested it.

5.8 Two Claim Chains

The two larger cases can now be expressed as different claim chains.

For the Superapp:

- more integrated functions
- broader functional coverage
- deeper architectural integration
- claims about resilience, independence or infrastructural strength

For Superintelligence:

- better performance across more tasks
- broader capability

- stronger inference and interaction
- claims about knowledge, generality or eventual closure

Neither chain is invalid by definition. Both contain transitions that may be real and may eventually be established. Their usefulness lies in making those transitions visible.

The next two sections examine the cases separately before the argument returns to their common structure.

6. Case I: The Superapp

A Superapp is often described through functional breadth. Payment, communication, identity, shopping, mobility, documents and other services become accessible through one application or ecosystem. This can create substantial practical value. Fewer interfaces can reduce friction, shared services can simplify coordination, and integrated data can enable functions that would be difficult to provide through isolated applications.

The architectural question begins when functional integration is treated as evidence for properties that depend on how the integrated system is structured.

6.1 Functional Breadth Is a Real Property

A platform that provides ten well-integrated functions has greater functional breadth than one that provides two. Shared authentication, common navigation and coordinated data flows can make those functions easier to use together. There is no need to diminish this gain in order to examine its consequences.

Functional breadth can itself be strategically important. It can reduce transaction costs, improve continuity across services and create new combinations of capability. The mistake would be to assume that every other desirable architectural property grows automatically with it.

6.2 Integration Changes Relations

As functions become integrated, the relations between them can change. A payment function may begin to depend on a shared identity mechanism. Communication may use the same authorization state as document access. Shopping history may influence mobility offers. A common data model may connect services that previously maintained independent records.

These relations can create genuine new capabilities, but they can also create new dependency paths. The architectural transition therefore cannot be described by counting functions alone. It depends on what the functions now require from one another.

6.3 Necessary Coupling and Inherited Coupling Are Different

Some coupling is intrinsic to a function. A payment requires some relation to an account or settlement mechanism. Access control requires some relation to identity or authorization. Other coupling is inherited from the architecture.

A payment function may technically require identity, for example, without requiring the same identity provider, data model or application state as every other function in the system. This distinction matters for substitutability. If a function cannot be replaced because unrelated responsibilities have become dependent on the same implementation, the dependency is architectural rather than intrinsic to the function itself.

6.4 Centralization Can Improve or Weaken Resilience

Centralization has no single effect on resilience. Replacing several inconsistent authentication systems with one robust mechanism can reduce attack surface and operational complexity. Centralizing a fragile dependency, however, can increase the number of functions affected by its failure. The same structural move can therefore improve one form of resilience while weakening another.

Resilience cannot be inferred from centralization or decentralization as labels. It has to be examined through failure paths, recovery mechanisms and the dependencies created by the resulting architecture.

6.5 Convenience and Independence Are Different Properties

A deeply integrated application can be exceptionally convenient while remaining dependent on one provider, one identity mechanism, one data structure or one policy boundary. This does not make the integration undesirable. It means that convenience and independence answer different questions.

Convenience concerns how easily functions can be accessed and combined. Independence concerns which conditions have to remain in place for those functions to continue existing or to be replaced. One can increase while the other decreases.

6.6 Lock-In Can Emerge Without Being Designed

Vendor lock-in does not require an explicit intention to make exit difficult. It can emerge gradually as shared interfaces become proprietary conventions, data histories accumulate in one structure, authentication becomes common across services and external systems begin to depend on the resulting implementation.

Each individual integration decision may be reasonable, while the combined architecture becomes increasingly difficult to substitute. Lock-in is therefore better examined as a property of accumulated relations than as a claim about provider intent.

6.7 Independence Is Not Isolation

An independent function does not have to operate alone. Payment can depend on identity, identity on authoritative records, and a human-facing application on several underlying services. The issue is not whether relations exist, but whether those relations require unnecessary ownership, duplication or shared implementation.

A system can be highly connected and still preserve replaceability between responsibilities. It can also appear modular while several components remain dependent on the same hidden state or control mechanism. Independence is therefore compatible with composition.

6.8 The Black Box Problem Begins Before Failure

A highly integrated system can work extremely well while becoming progressively harder to inspect from the outside. If identity, authorization, records, inference and application logic are deeply combined, a user or external system may see the result without being able to recover which responsibility produced which part of it.

This does not yet make the architecture a black box in the stronger sense developed later in the paper, but it creates the conditions under which important distinctions can become difficult to

reconstruct. The problem can therefore begin before anything fails: successful operation can coexist with decreasing visibility into dependency and provenance.

6.9 The Superapp Claim Chain

The Superapp case can now be stated more precisely:

more functions

→ greater functional breadth

→ deeper integration

→ new dependency structure

→ possible new system-level properties

The first four transitions can be directly observed in an architecture. The final step depends on which property is being claimed. Greater resilience requires evidence about failure and recovery. Independence requires evidence about dependency and replacement. Infrastructural strength may require evidence about continuity, governance or portability.

Functional breadth can contribute to all of them. It does not substitute for establishing them. The Superapp therefore demonstrates a broader principle: **accumulation can change architecture, but the properties attributed to that architecture have to be established at the level where they arise.**

The second case begins from a different kind of accumulation: capability rather than functions.

7. Case II: Superintelligence

The Superintelligence case begins from a different form of accumulation. A system can become better across more tasks, use more tools, integrate more information, improve its own strategies and interact more effectively with its environment. These developments can produce capabilities that are qualitatively different from those of earlier systems.

The question is not whether such transformation is possible. It is what follows from the capabilities that have actually emerged.

7.1 Capability Can Become Relational

Intelligence is often discussed as though capability were contained entirely inside a system. Many important capabilities, however, depend on relations. A system can become more capable because it can identify relevant information, select an appropriate tool, request a measurement, conduct an experiment or interact with another system. Its effective capability can therefore increase even when the required information or function is not internally stored.

This matters for claims about generality. A system does not have to contain every answer in advance to operate across a broad range of problems. It may instead possess increasingly effective ways of obtaining what a particular problem requires. Capability can therefore reside partly in access, selection and interaction.

7.2 Prediction and Observation Remain Different Relations

A sufficiently capable system may predict an external state with extraordinary accuracy. That does not require the prediction to become an observation. Prediction relates available information to a state that remains unresolved or inaccessible at the time of inference. Observation creates a different relation through measurement, sensing or interaction.

The distinction is not between weak prediction and strong observation. A prediction may be far more reliable than a noisy observation. The distinction concerns how the claim is connected to the state. If the system later measures the predicted event, its basis changes because the relation changes. The prediction may have been valid before the observation; the observation establishes something different.

Preserving this distinction does not diminish intelligence. It allows the system to represent what it knows, how it knows it and what remains dependent on external state.

7.3 Better Inference Can Improve the Decision to Observe

Greater intelligence can change the relation between inference and observation without collapsing them. A system may infer that an additional measurement is unnecessary because the available information already supports the required decision. In another case, it may identify precisely which missing observation would reduce uncertainty most effectively.

This is a stronger capability than either indiscriminate measurement or indiscriminate inference. The improvement lies partly in deciding when another relation to the environment is required.

7.4 Tool Use Extends Capability Without Requiring Ownership

A system that can use external tools can perform tasks that exceed the functions implemented directly within it. It may call a calculator, query a database, control an instrument, execute code or delegate a specialized operation to another system. The resulting capability belongs to the functioning system even though not every contributing function is internally owned.

This is not a weakness in the concept of capability. Human and technological systems routinely operate through reliable external relations. The relevant issue is whether the relation is available, appropriate and sufficiently reliable for the claim being made.

7.5 Interaction Can Produce New Information

Some information cannot be obtained by reasoning over a fixed internal state alone. An experiment can create a new observation. A sensor can expose a current physical condition. A question can elicit information from another agent. An action can alter the environment and reveal how it responds.

In these cases, interaction is not merely retrieving information that was already internally available in another form. It participates in producing the informational state from which the next inference can proceed. This gives capability a temporal dimension: what the system can know next can depend on what it does now.

7.6 Greater Capability Can Make Uncertainty More Explicit

A stronger system does not necessarily represent less uncertainty at every moment. It may recognize that a previously confident answer depended on an unsupported assumption. It may distinguish several plausible explanations where a weaker system saw only one. It may identify an unresolved external state that had previously been ignored.

In such cases, greater capability increases explicit uncertainty because the system represents the problem more accurately. This is not a regression. The ability to preserve an unknown where the basis does not support a stronger claim is itself an epistemic capability.

7.7 Generality Does Not Require a Closed World

A system can operate across an expanding range of problems without requiring the entire relevant world to be represented internally or fixed in advance. Generality can include the ability to encounter unfamiliar states, identify missing information, construct new procedures and establish new relations to tools or environments.

This makes generality compatible with openness. A system can become more general by becoming better at dealing with what it does not yet contain.

7.8 Capability Can Reside in Relations

The strongest architecture is not necessarily the one that contains the most internally. A system that can reliably obtain information, use tools, verify results and interact with its environment may possess greater effective capability than one with a larger fixed internal store but weaker external relations.

This does not imply that internal capability is unimportant. It means that internal resources and external relations can contribute to capability in different ways. The Superintelligence case therefore shifts the focus from possession to access and from accumulation alone to the relations through which capability becomes effective.

7.9 The Superintelligence Claim Chain

The case can now be stated as a sequence:

better performance across more tasks
→ broader capability
→ stronger inference, tool use and interaction
→ access to a wider range of states and problems
→ possible new forms of general capability

Each transition can be real. A further transition is required, however, when broader capability becomes a claim about complete knowledge, unrestricted generality or an eventual endpoint. Those properties have to be established through the relations and conditions they require.

Greater intelligence can extend epistemic reach dramatically. It does not need to erase the difference between what can be inferred, what can be observed and what remains dependent on future interaction. The next question is what happens when the system's own activity changes what becomes accessible next.

8. Recursive Capability

Capability becomes more difficult to describe as a fixed quantity when the result of one action changes the conditions for the next. A system that learns, experiments, asks questions or acts in its environment does not merely process a static set of available information. Its activity can change the state from which later learning proceeds.

This recursive structure matters because the next relevant distinction may not yet be available from the current state.

8.1 Learning Can Change What Becomes Learnable

Learning does more than add another item to an existing store. A newly acquired distinction can make previously unnoticed differences visible. A concept can enable a new question. A

successful interaction can reveal which observation should occur next. A new tool can make a previously inaccessible state measurable.

The result of one learning step can therefore change the space of possible next steps. This does not require unlimited growth. It requires only that the sequence of learning is partly state-dependent.

8.2 Childhood Language Acquisition Makes the Structure Intuitive

Childhood language acquisition provides an intuitive example. A young child does not begin with a complete list of all future linguistic distinctions and then fill them in one by one. Interaction, correction, imitation and use contribute to the distinctions that become available next. What has already been learned changes what can be noticed, attempted and understood in later interaction.^{[1][2]}

The comparison is not intended to establish a general superiority of biological learning or a limit on artificial systems. It illustrates a narrower point: the amount already known and the process through which the next learnable distinction arises are different properties. Developmental and enactive approaches are useful here because they treat cognition as something that develops through continuing relations between an agent and its environment rather than as the accumulation of detached representations alone.^{[1][2]}

The technological relevance lies in the recursive structure, not in treating a child as a model of an AI system.

8.3 A Better Question Can Be a Capability Gain

Progress is not limited to producing better answers to fixed questions. A system can improve by identifying that the current question is poorly specified, separating several problems that had previously been treated as one, or discovering a variable that changes what needs to be asked.

This kind of progress can initially look like complication. The system may replace one apparently simple question with several more precise ones. Yet the expanded problem representation can support stronger later claims because distinctions that were previously hidden have become explicit.

8.4 Interaction Can Change the Next State

When a system acts on the environment, the next state may differ not only because new information was retrieved but because the action itself changed what exists. An experiment changes conditions in order to expose a relation. A robot moves an object and subsequently observes a different physical arrangement. A negotiation changes the state of the parties involved. A software system can modify the infrastructure it later has to inspect.

The next inference therefore begins from a state partly produced by the preceding action. This creates a feedback relation between capability, action and observation.

8.5 Recursive Capability Changes the Meaning of Reach

If each successful step can expose new distinctions, tools, observations or questions, epistemic reach cannot always be understood as movement across a fully specified map. The system may become able to address states that were previously inaccessible, but it may also become able to formulate distinctions that were not previously operationally relevant.

Greater reach can therefore reveal additional unknowns rather than simply reduce a fixed remainder. This does not make progress indeterminate. Local problems can still be solved, measurements can become more accurate and uncertainty can fall within defined questions. The recursive point concerns what becomes available as a relevant next question.

8.6 Better Questions Can Increase Explicit Uncertainty

A system that discovers a previously ignored distinction can become less certain about a claim it once made confidently. Suppose a model initially treats two operating conditions as equivalent. Later interaction reveals that they produce systematically different outcomes. The improved representation now contains an additional variable and may assign greater uncertainty until enough evidence has been gathered under both conditions.

Capability has increased even though confidence has temporarily fallen. The system is now uncertain about something it previously failed to distinguish.

8.7 Recursion Does Not Guarantee Unlimited Development

Recursive learning should not be turned into the opposite overclaim. The fact that one learning step can create new opportunities for another does not establish that the process can continue without bound. Physical constraints, computational limits, unavailable observations, irreducible uncertainty or the structure of a particular problem may all matter.

Recursion establishes that the next relevant state can depend on the path taken to reach it. It does not establish the ultimate extent of that path.

8.8 The Remaining Question Is About the Horizon

Recursive capability changes the closure question. If a system is moving through a fully specified problem space, progress can in principle be measured against what remains. If its interactions also expose new distinctions and new relevant questions, the boundary against which progress is evaluated can change with the process.

That boundary is the next issue. It is useful to think of it as a horizon: not necessarily a final limit, but a boundary relative to a current position and means of access.

9. The Moving Horizon

A horizon is not simply a limit. It is a boundary relative to a current position and means of access. This makes it useful for recursive capability. A system can extend what it can observe, infer or act upon while simultaneously changing which distinctions become relevant from its new position. Progress and an expanding horizon can therefore coexist.

9.1 A Horizon Is Relative to Position

A horizon depends on where observation occurs and what forms of access are available. A new sensor can expose states that were previously inaccessible. A new concept can make a previously unnoticed distinction operationally relevant. A new tool can allow a system to interact with an environment in a way that creates different observations.

The horizon moves because the relation between the system and what it can access has changed. This is different from saying that the underlying world must have expanded. What changed may be the system's position, capability or means of access.

9.2 Extending Reach Can Reveal More Beyond It

Greater capability does not guarantee that the unresolved remainder will appear smaller from every new position. A better instrument can reveal finer structure. A broader model can expose variables that an earlier representation ignored. A successful experiment can generate several new questions. A system that becomes better at identifying its own unsupported assumptions may represent more explicit unknowns than before.

This is compatible with real progress. The system can know more and distinguish more while also becoming able to represent more of what it does not yet know.

9.3 A Cosmological Analogy

A limited analogy can make the structure intuitive. In cosmology, expansion is not ordinarily described as objects simply moving toward a fixed outer wall. Spatial relations themselves can change, so the intuitive picture of a traveller approaching a static edge is misleading.^[3]

The epistemic parallel is narrow. A system extending its capabilities does not have to be understood as moving across a fixed map toward a visible final boundary. Its new position can change what becomes accessible and which distinctions become relevant. Knowledge is not physical space, intelligence is not a cosmological object, and cosmological expansion establishes nothing about the future of AI. The analogy illustrates only why progress does not require a static horizon.

9.4 A Moving Horizon Does Not Mean No Progress

If the horizon changes, progress can still be measured locally. A system can solve a defined problem, reduce uncertainty about a specified state, improve prediction, verify a property or extend reliable operation into new conditions. These gains remain real even if new questions become visible afterwards.

Progress therefore does not require the total set of possible future questions to shrink monotonically. It requires that something identifiable has been established relative to the problem being addressed.

9.5 Local Closure and General Closure Are Different

Some problem spaces are sufficiently bounded for completion to be meaningful. A finite search can terminate. A proof can establish a proposition within a formal system. A verification procedure can satisfy defined conditions. An experiment can resolve a specified empirical question to an accepted level of uncertainty. These are forms of local closure.

A much stronger claim is that the relevant space as a whole has been exhausted or that no further distinction could matter to the broader property being claimed. Success at local closure does not establish general closure.

9.6 An Expanding Horizon Is Not an Infinite Horizon

The horizon analogy should not be converted into another endpoint claim. The fact that new distinctions can become visible does not establish that this process continues indefinitely. There may be limits imposed by the world, the architecture, available energy, computation or the structure of the problems themselves.

A moving horizon establishes only that the current boundary cannot automatically be treated as a fixed final boundary. Whether a final boundary exists is a separate question.

9.7 A Fixed Horizon Would Still Not Settle the Endpoint

Even if the relevant space were fixed and finite, continued improvement would not by itself establish which qualitative state appears at the end of the trajectory. Knowing that a process has a boundary is different from knowing the properties of the system that reaches it.

This separates two questions that can otherwise become entangled: Is the relevant space fixed? And if it is, what follows from continued movement within it? The second question remains even when the first is answered.

9.8 From Horizon to Continuation

The horizon argument concerns the structure of the space through which development proceeds. It asks whether the relevant boundary can change as capability changes. A different problem remains if the space is assumed to be stable.

Even then, continued improvement does not automatically establish completion or determine the qualitative nature of an endpoint. That is the next step.

10. Continuation Is Not Completion

Assume, for the moment, a stable reference space. A system improves repeatedly within it. Error falls, performance rises or an approximation becomes increasingly precise. The direction of development is clear. The remaining question is whether continuation itself establishes arrival.

A simple mathematical analogy isolates this question. Successive approximations of π can become arbitrarily accurate while remaining approximations. The example is useful not because intelligence resembles π , but because it separates continued improvement from the additional claim that the process has reached a particular endpoint.

10.1 What the Analogy Establishes

The analogy makes only a narrow point. A sequence can improve consistently. Each later state can be better than the previous one by a defined measure. The direction can be stable and the remaining error can become extremely small. None of this, by itself, turns continuation into completion.

This says nothing about an ultimate limit on intelligence. It does not show that AGI, Superintelligence or any other proposed technological state is unreachable. It shows only that the existence of an improving sequence is not sufficient evidence for every endpoint attributed to that sequence.

10.2 Practical Sufficiency Is Not the Same as Exact Completion

In practice, exact completion is often unnecessary. A numerical approximation can be sufficient for engineering. A prediction can be sufficiently accurate for a decision. A model can perform well enough for a defined task. A system can satisfy an operational requirement without eliminating every remaining uncertainty.

This is genuine success, and the distinction matters only when practical sufficiency is converted into a stronger claim about completion. A system can be adequate for its purpose without the broader reference space being exhausted.

10.3 Convergence Does Not Establish Arrival

A process can converge toward a limit without the observed trajectory establishing that the limit has been reached. Technological development can show analogous patterns. Performance may improve across successive evaluations, error rates may fall, or capability may expand across an increasing range of tasks.

Such convergence can support expectations about further progress. What it establishes depends on the measurements and conditions through which the trajectory is observed. Arrival remains a separate claim.

10.4 Closure Is a Stronger Claim

Closure means more than high performance or low uncertainty. Within the relevant frame, nothing required by the claim remains unresolved. That can be established for bounded problems: a finite search can be exhausted, a formal proof completed or specified verification conditions satisfied.

For broader claims, closure requires a basis appropriate to their scope. Strong performance across many tasks does not by itself establish that no relevant capability remains outside the tested set. Extremely reliable inference does not establish that no unresolved external state could matter. Closure therefore depends on the boundary of the claim, not simply on how far a trajectory has progressed.

10.5 Qualitative Transition Remains Possible

None of this rules out genuine qualitative transition. Water shows that accumulation can participate in a real change of state. A technological system may likewise cross a threshold, acquire a new architecture or develop a capability that was absent before.

The point is narrower: continuation alone does not identify which transition has occurred. Once a new property can be established in the resulting system, the description can change with it. The transition is then supported by the properties of the new state rather than by extrapolation from the trajectory alone.

10.6 The Endpoint Cannot Explain the Path Backwards

Once an endpoint has been assumed, earlier development can easily be reinterpreted as though every preceding step were already evidence for that destination. This reverses the direction of support.

A trajectory can justify expectations about what may follow. A later state, once actually reached, can also change how the history of that trajectory is understood. But an anticipated endpoint cannot serve as evidence that its own preconditions have already been established. The path can point toward a hypothesis without borrowing validity from the hypothesis itself.

10.7 Stable Space and Moving Horizon Are Different Problems

The previous section asked whether the relevant horizon can change as capability changes. This section assumed the opposite condition: a stable reference space. The two cases therefore test different things.

If the horizon moves, the relevant accessible space cannot simply be treated as a fixed remainder waiting to be exhausted. If the reference space is stable, continued improvement within it still does not establish completion by itself. Neither case decides what future

technological endpoint will exist. Together they show why the structure of the space and the trajectory through it have to be kept distinct.

10.8 What Continuation Can Establish

Continuation can establish a great deal. It can establish persistent improvement, increasing accuracy, broader capability, decreasing error or convergence under specified conditions. It can justify extrapolation and motivate hypotheses about future states.

What it cannot supply on its own is the additional basis required for a qualitatively different claim. The relevant issue is therefore not whether development can continue far beyond current capability, but what becomes established as that development proceeds. This returns the argument to the two boundaries introduced at the beginning: how far the claim extends, and how far its basis extends with it.

11. Claim Reach and Validity Reach

The preceding sections have approached the same problem through different forms of transformation. A property can change. A claim can move between levels. A relation can change. Conditions can disappear. Several individually plausible transitions can accumulate into a broader conclusion. Capability can also change what becomes accessible next.

The distinction introduced at the beginning can now be stated more precisely. **Claim Reach** describes how far a statement extends: which property it asserts, at what level, through which relation and under which conditions. **Validity Reach** describes how far the basis for that statement supports it. The two boundaries can move together, but they do not have to.

11.1 Reach Is About Boundaries, Not Scores

Claim Reach and Validity Reach are not numerical measures of truth. A narrow claim is not inherently better than a broad one, and a broad claim is not inherently less valid. A highly ambitious claim can be fully established if its basis supports what it says.

Reach describes boundaries. A claim exceeds its current Validity Reach when it asserts a property, level, relation or set of conditions that its basis does not yet establish. This can be temporary because the basis can grow.

11.2 Validity Reach Can Grow

New observations can extend an empirical basis. Experiments can test additional conditions. Formal analysis can establish structural properties. Operational history can expose behavior across time. New interactions can make previously unresolved states accessible.

Validity Reach therefore changes as the relations supporting a claim change. A statement that was initially speculative can later become well established without changing its wording because the basis available to support it has changed. This is why the distinction is not a rule against ambitious claims; it preserves the difference between what is proposed and what has already been established.

11.3 Better Relations Can Extend Reach

More evidence is not the only way to extend Validity Reach. A better relation to the relevant state can matter just as much. A prediction can later be supplemented by observation. A component result can be extended by system-level testing. A theoretical dependency can be examined

through actual replacement. A claim about behavior under one condition can be tested under additional conditions.

The basis grows because access to what matters for the claim has changed.

11.4 Different Boundaries Can Move at Different Speeds

Technological development can extend Claim Reach and Validity Reach at different rates. Language may expand quickly because a trajectory strongly suggests a broader future state. Evidence may expand more slowly because the new property requires additional observation, testing or operation.

The reverse can also happen. New evidence may support a broader statement before the terminology used to describe the system changes. Neither sequence is inherently problematic; the important point is that movement on one side should not be mistaken for movement on the other.

11.5 Claim Reach Can Also Shrink

A claim does not always become more valid by acquiring more evidence. It can also become more valid by becoming more precise.

“Reliable” can become “reliable under these operating conditions.”

“General” can become “general across these evaluated task classes.”

“Independent” can become “replaceable at this interface while retaining these dependencies.”

The narrower statement may communicate more because its boundary is clearer. Reducing Claim Reach can therefore be an epistemic improvement.

11.6 Unknown Is a Valid State

When the available basis does not establish a stronger claim, the correct state does not have to be a weaker guess. It can be unknown.

This matters in systems that are expected to reason under uncertainty. Preserving an unresolved state can prevent prediction from being represented as observation, expectation as verification or extrapolation as established endpoint. Unknown is not the absence of intelligence; it is a valid representation of the current boundary of support.

11.7 Reach Changes Over Time

The relation between Claim Reach and Validity Reach is historical. A claim can begin as a hypothesis, acquire experimental support, accumulate operational evidence and later become established across a broader range of conditions. Another claim can lose support when the environment changes or a previously hidden dependency becomes relevant.

Validity therefore cannot always be understood from the current statement alone. Its basis and the path through which that basis developed may matter.

11.8 The Relation

The relation can now be read without treating it as mathematics:

Claim Reach $\leq$ Validity Reach

A claim should not extend beyond the basis that currently supports it. This does not impose a permanent limit on what can be claimed. Both boundaries can move. New evidence, new observations, new relations and new system properties can extend what is validly sayable.

The relation is therefore dynamic. It does not ask technological development to remain inside today's boundaries. It asks that the boundary of the claim remain distinguishable from the boundary of its current basis. That distinction becomes architectural when several different kinds of state, evidence and interpretation participate in one system.

12. Separation Without Fragmentation

Recording, inference, prediction, validation, contextualization and interaction can participate in one system while depending on different conditions of validity. Keeping those responsibilities distinguishable does not require building isolated systems around each of them.

The architectural problem is therefore not how to separate everything, but how to compose different responsibilities without making their boundaries unrecoverable.

12.1 Separation Is About Responsibility

Two functions can interact continuously and still remain distinguishable. A measurement can feed an inference process. An inference can trigger another observation. A contextual representation can combine information from several sources. None of these relations requires the participating responsibilities to become the same thing.

Separation therefore concerns responsibility rather than absence of connection. The system should retain enough structure to recover what produced a state, what transformed it and what later interpretation was added to it.

12.2 Composition Does Not Require Duplication

One responsibility does not have to copy another in order to use it. A persistent record can support later reasoning without containing that reasoning. A contextual representation can combine information from several sources without becoming the canonical origin of each source. A semantic reference can be reused without being redefined wherever it appears.

The important distinction is between reuse through relation and duplication through redefinition. Relations allow information to participate in several contexts while preserving where it came from.

12.3 Canonical Identity Reduces Ambiguity

When the same concept, entity or record is represented independently in several places, later changes can create competing versions. A canonical reference provides a stable point to which other representations can relate.

This does not require every use of the information to look the same. Human-facing explanations, machine-readable structures and operational records can serve different purposes while retaining a recoverable relation to what they reference. Canonical identity therefore supports variation without requiring redefinition.

12.4 Human and Machine Requirements Can Differ

A representation optimized for human understanding does not have to be identical to one optimized for machine processing. Humans may need narrative, examples, visual hierarchy and

contextual explanation. Machines may need stable identifiers, explicit relations, structured state and provenance.

Trying to make one representation perform every role can weaken both. The alternative is not duplication. Different representations can be composed from shared references and records while preserving their distinct purposes.

12.5 Provenance Preserves the Path

When a later claim depends on several transformations, the current state alone may not show how it arose. Provenance can preserve which observation, record, inference or contextual step contributed to the result. This becomes increasingly important when systems update recursively or combine information from several sources.

The purpose is not to record every internal operation indefinitely. It is to preserve the transformations that matter for reconstructing the basis of a relevant claim. Provenance therefore connects architecture to validity.

12.6 Separation Makes Replacement Observable

Substitutability is easier to evaluate when responsibilities remain distinguishable. If replacing one component requires rewriting unrelated concepts, migrating hidden shared state and reconstructing contextual representations, the architecture has inherited dependencies that may not be visible at the interface.

When responsibilities relate through explicit boundaries, replacement can be examined more directly. Separation does not guarantee substitutability, but it makes the relevant dependencies easier to observe.

12.7 One Concrete Realization

The general principle can be expressed more concretely through three distinguishable responsibilities used in related work: a Semantic Core (SC), a Persistent Record Layer (PRL) and a Human Context Layer (HCL).^[4] These responsibilities are distinguishable but not structurally symmetrical: contextual representation depends on an established semantic reference for the subject it represents.

The Semantic Core provides stable semantic references and relations. The Persistent Record Layer preserves observations, states, identity and provenance across time. The Human Context Layer provides contextual representations for human use in relation to an established semantic reference, without becoming the canonical origin of the underlying records or semantic definitions.

These responsibilities are related but not interchangeable. They do not require three isolated systems, nor do they require copies of the same information in each place. This is one concrete realization of the broader principle, not a universal requirement for every system.

12.8 Composition Can Be Recursive

A contextual representation can refer to a semantic concept and a recorded state. A later interaction can create a new record. That record can support another inference, which can in turn change what context becomes relevant next.

The resulting system is recursive without requiring each responsibility to absorb the others. New states can be added, new supported relations can be established within the responsibility to

which they belong, and new contexts produced while earlier records and references remain recoverable. This allows development without requiring every new step to rewrite the structures from which it emerged.

12.9 Agnosticism Preserves Replaceability

A responsibility becomes easier to replace when it depends on what another responsibility provides rather than on how that responsibility is internally implemented. A contextual responsibility can consume records without owning their storage model. A record responsibility can preserve states without determining how they must later be explained. A semantic responsibility can provide stable references without prescribing every application that uses them.

This kind of agnosticism does not remove dependency. It makes the dependency more explicit and narrower. Replaceability then becomes a property of the relation rather than an assumption based on nominal modularity.

12.10 Separation Supports Change

A system that preserves distinctions between semantic reference, recorded state and contextual interpretation can change each responsibility at a different rate. A new context does not require rewriting history. A new record does not require redefining the concept it concerns. A revised semantic relation does not require pretending that earlier observations were recorded under the new interpretation.

This supports both continuity and change. The architecture remains compositional because the responsibilities can interact, and it remains traceable because their roles do not have to collapse into one another.

Composition does not require collapse.

13. What Accumulation Does Not Decide

The two cases began with different forms of accumulation. The Superapp accumulated functions; Superintelligence accumulated and combined capabilities. Both can produce real transformation. What accumulation does not decide on its own is which additional property has been established. That depends on the resulting structure and the relations through which the property exists.

13.1 Accumulation Does Not Name the Result

More functions establish greater functional breadth. Better performance establishes improvement across the conditions in which it was measured. Neither result is trivial. A further step is required when the description changes category: breadth becomes independence, performance becomes generality, prediction becomes knowledge or continued development becomes completion.

The later property may be real. It has to be established as that property rather than inherited from the direction of the preceding change.

13.2 Some Boundaries Carry Information

Boundaries are not only restrictions. Some preserve distinctions that the system needs in order to represent what happened. Separating observation from inference preserves how a state

became known. Separating recorded state from later interpretation preserves history. Separating responsibilities in an architecture can expose which dependency belongs to which function.

Removing such boundaries can therefore remove information. A system does not become more integrated merely because every distinction disappears. In some cases, the distinction is part of what makes reliable integration possible.

13.3 Epistemic Differentiation Can Be a Capability

A capable system can distinguish between what it observed, what it inferred, what it predicted and what remains unresolved. That differentiation does not weaken the system's knowledge; it makes the basis of different claims recoverable.

Greater intelligence therefore does not require observation, inference, prediction and unresolved external state to collapse into one category. It can instead improve the system's ability to decide which relation is appropriate and when another observation or interaction is required. Knowing the boundary of a claim can itself be part of capability.

13.4 Independence Is Relational

Independence does not mean absence of dependency. A function can depend on identity, data, infrastructure or another service while remaining replaceable within those relations. The relevant issue is whether its operation depends on unnecessary ownership, duplication or implementation-specific coupling. Independence is therefore better understood through the structure of dependencies than through isolation.

13.5 Relations Can Be a Source of Strength

A relation is not necessarily a weakness. Reliable access to an external observation can be stronger than an internal estimate. A well-defined interface can be more robust than duplicated functionality. An authoritative external record can be preferable to a local copy that gradually diverges from its source.

The strength of a system can therefore depend on the quality of its relations as well as on the capabilities contained within it. This changes the meaning of self-sufficiency: a system does not have to own every resource it can use effectively.

13.6 Capability Need Not Be Contained Internally

A system can extend its effective capability through reliable access to information, tools and other services. What matters is not simply whether a capability resides inside or outside the system, but whether the relation through which it is used is appropriate to the claim being made.

Internal capability and external relations can therefore complement one another. Strength need not be measured by how much the system contains.

13.7 Accumulation Remains Powerful

None of this diminishes accumulation. More data can improve models. More functions can create useful platforms. More computation can enable new capabilities. More observations can extend a basis. More experience can expose structures that were previously invisible.

Accumulation becomes problematic only when the property attributed to the result is treated as though it were already contained in the quantity that preceded it. The stronger position is not to reject extrapolation, emergence or technological ambition, but to keep the transition visible.

When the resulting property becomes observable, testable, verifiable or otherwise supportable through an appropriate relation, the claim can change with it. The remaining complication is that recursive development can alter more than the system itself. It can also change the conditions under which the next state emerges.

14. When the Process Changes the Problem

So far, transformation has usually been considered as a change in the system or in the reach of a claim about it. Recursive systems introduce a further complication. A successful action can change the environment in which the next action occurs. A new capability can make previously irrelevant distinctions important. An architecture can create dependencies that later become part of the conditions under which it has to operate.

The process is no longer only moving through a problem. It can participate in changing the problem itself.

14.1 A Solution Can Create a New Problem State

A solution does not always leave the original problem space unchanged. Automation can remove one operational constraint while creating new dependencies on infrastructure, supervision or recovery mechanisms. A platform can simplify coordination while creating questions of portability or institutional dependence that were less relevant before integration. An AI system can solve a class of tasks and thereby make a different class of questions operationally accessible.

The original problem may have been solved within its earlier frame. What follows is not necessarily evidence that the solution failed. The success itself can create a new state in which different properties become relevant. Development can therefore transform the conditions against which its next stage is evaluated.

14.2 Observation Can Be Part of the Transition

This becomes particularly important when the system observing a state also acts within the environment from which that state emerges. An experiment changes conditions in order to make a relation observable. A robot alters its physical environment and subsequently observes the changed state. A platform influences behavior that later appears in its operational data. An adaptive system can select actions partly because of what it previously observed.

This does not mean that observation creates reality in a general sense. The narrower point is that the route through which a later state becomes observable can depend on earlier actions and distinctions. Observation is therefore sometimes part of the history of the transition rather than an external view of an otherwise unchanged process.

14.3 The Reference Space Can Develop With the System

If new capabilities create new actions, observations and relevant distinctions, the reference space against which development is evaluated need not remain fully specified in advance. A capability that does not yet exist may have failure modes that cannot yet be observed operationally. A new instrument can make distinctions measurable that previously had no practical role. A new architecture can create relations whose consequences become visible only after adoption.

The relevant space is therefore not merely unknown in the sense of containing a fixed set of unanswered questions. Parts of its structure can become operationally relevant through development itself. This is stronger than saying that the horizon moves. A moving horizon concerns what becomes accessible from a changing position. Here, the development can also alter which relations, dependencies or distinctions constitute the problem being examined.

14.4 Difficulty Is Relative to Capability

A problem that is difficult for one system can become routine for another. Human cognitive limits therefore do not define a permanent upper boundary for what a more capable system could solve. Future systems may resolve problems that currently require substantial expertise or cannot presently be solved at all.

But removing one boundary does not establish the absence of every boundary. New capability can expose more complex interactions, finer distinctions or entirely different questions. Difficulty is therefore relational: it depends partly on the capability available to address the problem. This allows technological development to be taken seriously without turning present human limitations into universal limits or future capability into automatic closure.

14.5 Architecture Changes Its Own Operating Environment

The same recursive structure appears in integrated systems. Once an architecture is adopted, organizations adapt processes around it. Interfaces become conventions. Data histories accumulate. Other services integrate with its mechanisms. Legal and institutional practices can begin to assume its continued existence.

A component that was initially easy to replace may therefore become harder to replace later even if its own technical implementation has not changed. The relevant dependency is no longer contained only in the component; it has developed through relations between the architecture and its environment.

This is why substitutability and independence can have a historical dimension. The current structure alone may not reveal how the present dependencies arose.

14.6 Recursive Systems Need History

If the current state depends partly on previous interactions, two apparently similar states can have different meanings because they arrived there through different transformations. The same value may have been observed directly in one case and inferred in another. The same configuration may result from deliberate design or from accumulated compatibility constraints. Two systems may expose similar interfaces while one remains readily replaceable and the other carries years of inherited dependency.

A purely current-state representation can hide these differences. History therefore has a role beyond storage. It allows transformations to remain traceable: what changed, what produced the change and which later claims depend on that path. For recursive systems, provenance is not merely descriptive metadata added after the fact. It can be part of the basis required to understand the present state.

14.7 When Transformation Becomes a Black Box

A system can be complex without being a black box in the sense relevant here. Complexity becomes an epistemic problem when the transformations needed to reconstruct a claim are no longer distinguishable. If observation, inference, contextual interpretation and later action

collapse into an opaque current state, it can become difficult to determine which part of that state was observed, which was derived and which changed through subsequent interaction.

The same problem can occur architecturally. An integrated platform may continue to function while the dependencies through which its current state developed become difficult to recover. A replacement can then fail for reasons that were invisible in the apparent structure of the component being replaced.

Opacity in this sense is not simply complexity. It is the loss of distinctions required to reconstruct relevant transformation.

This also distinguishes system-level traceability from the narrower question of whether the internal reasoning of an individual model can be explained. A model may be interpretable while the provenance of the surrounding system is opaque, or the reverse. The Black Box problem here is therefore relational and historical: can the path from basis through transformation to current claim still be reconstructed where that path matters?

14.8 Context Can Change Without Rewriting History

Recursive development also means that the relevance of an earlier state can change later. A record created for one operational purpose may become relevant to a later investigation. A technical event may acquire legal or organizational significance that was not present when it occurred. New knowledge can alter how an earlier observation is interpreted.

The earlier state does not have to be rewritten for this to happen. Keeping persistent state distinguishable from later context allows new interpretations to develop while preserving what was originally recorded. Semantic distinctions can likewise evolve when new relations become necessary without requiring every historical state to be retroactively redefined.

There may therefore be no final version of context even when the underlying record remains stable enough to preserve continuity.

14.9 No Fixed Problem Space, No Automatic Final State

Bounded completion remains possible throughout this argument. Algorithms can terminate, searches can exhaust defined spaces, verification conditions can be satisfied and specified questions can be resolved. The stronger claim concerns open-ended recursive development.

If a process can change its environment, expose new distinctions, create new dependencies and alter which questions become relevant, continued success does not by itself define a final state for the entire process. Such an endpoint may exist. A sufficiently capable future system may resolve classes of problems far beyond anything currently accessible. But the endpoint would require its own basis; it cannot be established merely from the fact that earlier development continued successfully.

History and traceability matter precisely because the path can change what later claims mean. When the relevant transformations become opaque, not only the current claim but also the boundary of its basis becomes harder to reconstruct.

The final question is therefore no longer whether transformation can occur. It clearly can. It is what should be preserved when systems, claims and the spaces around them continue to change.

15. Conclusion: When More Becomes Different

Technological development can produce genuine transformation. More functions can become a different architecture. More capability can produce properties that did not previously exist. Better relations to information, tools and environments can extend what a system is able to establish. Nothing in this paper requires technological progress to remain quantitative.

The difficulty begins when the direction of that progress is treated as sufficient evidence for every property attributed to its future state. A trajectory can establish improvement without defining its destination. Integration can establish functional breadth without determining resilience or independence. Greater intelligence can extend epistemic reach without making inference, observation and unresolved external state indistinguishable.

The relation can therefore be stated once more:

Claim Reach $\leq$ Validity Reach

This is not a fixed limit on what may eventually become true. Both boundaries can move. New observations, experiments, verification, operational history and better relations can extend Validity Reach, while newly established properties can support broader claims. The relation permits technological ambition, hypothesis and extrapolation while keeping visible the difference between what has been established and what is being projected.

This becomes especially important in recursive systems. Development can change not only capability but also the relations, dependencies and questions that shape what happens next. History matters because the current state may no longer reveal how it arose.

Opacity in this sense is not simply complexity. It is the loss of distinctions required to reconstruct relevant transformation.

Preserving those distinctions does not require fragmentation. A coherent system can combine responsibilities, information and capabilities while keeping the relations necessary for validity, provenance and change recoverable. Independence need not mean isolation, and strength need not mean containing everything internally.

This leaves technological ambition intact. Future systems may cross thresholds that are difficult to specify today, acquire capabilities that are qualitatively different from current ones and establish claims that presently remain hypotheses. If the basis changes, the valid claim can change with it.

What matters is not whether a predefined amount has been reached but whether the claimed difference has become a property of the system. More becomes different when the difference no longer depends on extrapolating the trajectory that preceded it, but can be established in the structure, behavior, relations or capabilities of what now exists.

And when that happens, the claim is allowed to move with it.

Resources

[1] **Jean Piaget.** *The Origins of Intelligence in Children.* Developmental reference for the relation between learning, interaction and the emergence of subsequent cognitive distinctions.

[2] Francisco J. Varela, Evan Thompson and Eleanor Rosch. *The Embodied Mind: Cognitive Science and Human Experience*. Reference for the enactive perspective on cognition as an ongoing relation between an agent and its environment.

[3] NASA Science. *Universe and cosmic expansion — reference material*. Factual reference for the limited cosmological analogy used in Section 9.3.

[4] Related architecture work.

Semantic Core (SC) — <https://www.robotasaservice.com>

Persistent Record Layer (PRL) — <https://www.carbontracking.de/method/>

Human Context Layer (HCL) — <https://context.robotasaservice.com>

Rights of Robots — <https://www.rightsofrobots.com>

Theoretical Context

Rights of Robots — Theoretical Context. Selected comparative context for distinction, representation, interpretation, organization, autonomy, stability and change, including Spencer-Brown, Varela and Luhmann.